What is claimed is:

1. An apparatus for reducing cache memory misses in a computer that performs context switches between at least a first context and a second context, the apparatus comprising:

a first logic, the first logic being configured to identify at least a first prefetch region in a first memory element during compilation of a computer program by the computer;

a second logic, the second logic being configured to identify critical memory references within the first prefetch region during compilation, the critical memory references within the first prefetch region corresponding to data that may be needed in cache memory if a context switch occurs from a process or thread associated with the second context to a process or thread associated with the first context during program execution by the computer; and

a third logic, the third logic being configured to prefetch data associated with the identified critical memory references and to store the prefetched data in cache memory prior to a process or thread associated with the first context is resumed when a switch from the second context to the first context occurs during program execution.

2. The apparatus of claim 1, wherein the third logic comprises logic configured to generate prefetch code during said compilation, the prefetch code being used by the third logic to prefetch the data associated with the identified critical memory references.

3. The apparatus of claim 1, wherein the third logic comprises logic configured to generate prefetch code and to store the prefetch code at a particular location in a memory element during compilation of a computer program.

4. The apparatus of claim 3, wherein the third logic comprises logic configured to store a location of the prefetch code, the third logic storing the location of the prefetch code during program execution, and wherein, prior to resuming execution of the process or thread associated with the first context, an operating system of the computer determines the location at which the prefetch code is stored and executes the prefetch code thereby obtaining the data to be prefetched and storing the prefetched data in cache memory.

5. The apparatus of claim 1, wherein the third logic comprises logic configured to generate code to store a prefetch count at a particular location in a memory element, the prefetch count corresponding to a number of registers needed for storing addresses at which data corresponding to the critical memory references is stored in a memory element.

6. The apparatus of claim 5, wherein the third logic comprises logic configured to allocate a prefetch memory register to each critical memory reference, and wherein a first one of the prefetch memory registers is a constant starting register.

7. The apparatus of claim 6, wherein, after a context switch from a process or thread associated with the first context to a process or thread associated with the second context has occurred, and prior to resumption of the process or thread associated with the first context has occurred, an operating system of the computer obtains the prefetch count and obtains the data stored at the addresses comprised in the prefetch memory registers beginning with the constant starting register and continuing sequentially through a number of the allocated prefetch memory registers dictated by the prefetch count.

8. A method for reducing cache memory misses in a computer that performs context switches between at least a first context and a second context, the method comprising the steps of:

identifying at least a first prefetch region in a first memory element during compilation of a computer program by the computer;

identifying critical memory references within the first prefetch region during the compilation, the critical memory references within the first prefetch region corresponding to data that may be needed in cache memory if a context switch occurs from a process or thread associated with the second context to a process or thread associated with the first context during program execution by the computer; and

during program execution, prefetching data associated with the identified critical memory references and storing the prefetched data in cache memory prior to the process or thread associated with the first context being resumed when a switch from the second context to the first context occurs.

9. The method of claim 8, further comprising the step of:

prior to prefetching the data, generating prefetch code during said compilation, the prefetch code being used during program execution to prefetch the data associated with the identified critical memory references.

10. The method of claim 8, further comprising the step of:

prior to prefetching the data, generating prefetch code during said compilation, the prefetch code being used during program execution to prefetch the data associated with the identified critical memory references; and

after generating the prefetch code, storing the prefetch code at a particular location in a memory element during compilation of a computer program.

11. The method of claim 10, further comprising the step of:

after generating the prefetch code, storing a location of the prefetch code in a memory element during program execution; and

prior to resuming execution of the process or thread associated with the first context, determining the location at which the prefetch code is stored and executing the prefetch code, thereby obtaining the data to be prefetched and storing the prefetched data in cache memory.

12. The method of claim 8, further comprising the step of:

prior to prefetching the data, generating code to store a prefetch count at a particular location in a memory element, the prefetch count corresponding to a number of registers needed for storing addresses at which data corresponding to the critical memory references is stored in a memory element.

13. The method of claim 12, further comprising the steps of:

allocating a prefetch memory register to each critical memory reference, and wherein a first one of the prefetch memory registers is a constant starting register.

14. The method of claim 12, further comprising the steps of:

after a context switch from a process or thread associated with the first context to a process or thread associated with the second context has occurred, and prior to resumption of the process or thread associated with the first context has occurred, obtaining the prefetch count and obtaining the data stored at the addresses comprised in the prefetch memory registers beginning with the constant starting register and continuing sequentially through a number of the allocated prefetch memory registers dictated by the prefetch count.

15. A computer program for reducing cache memory misses in a computer that performs context switches between at least a first context and a second context, the computer program being embodied on a computer readable medium, the computer program comprising:

a first code segment for identifying at least a first prefetch region in a first memory element during compilation of a computer program by the computer;

a second code segment for identifying critical memory references within the first prefetch region during the compilation, the critical memory references within the first prefetch region corresponding to data that may be needed in cache memory if a context switch occurs from a process or thread associated with the second context to a process or thread associated with the first context during program execution by the computer; and

a third code segment for prefetching data associated with the identified critical memory references and storing the prefetched data in cache memory prior to the process or thread associated with the first context being resumed when a switch from the second context to the first context occurs.

16. The computer program of claim 15, further comprising:

a fourth code segment for generating prefetch code during said compilation, the prefetch code being used during program execution by the third code segment to prefetch the data associated with the identified critical memory references.

17. The computer program of claim 15, wherein the third code segment further comprises:

a code segment for generating prefetch code during said compilation, the prefetch code being used during program execution by the third code segment to prefetch the data associated with the identified critical memory references and to store the prefetch code at a particular location in a memory element during compilation of a computer program.

18. The computer program of claim 17, wherein the third code segment further comprises:

a code segment for storing a location of the prefetch code in a memory element during program execution; and

a code segment for determining the location at which the prefetch code is stored and executing the prefetch code, thereby obtaining the data to be prefetched and storing the prefetched data in cache memory.

19. The computer program of claim 15, wherein the third code segment further comprises:

a code segment for generating code to store a prefetch count at a particular location in a memory element, the prefetch count corresponding to a number of registers needed for storing addresses at which data corresponding to the critical memory references is stored in a memory element.

20. The computer program of claim 19, wherein the third code segment further comprises:

a code segment for allocating a prefetch memory register to each critical memory reference, and wherein a first one of the prefetch memory registers is a constant starting register; and

a code segment for obtaining the prefetch count and obtaining the data stored at the addresses comprised in the prefetch memory registers beginning with the constant starting register and continuing sequentially through a number of the allocated prefetch memory registers dictated by the prefetch count.